

REMARKS

Claims 1-20 are pending in the current application. In an Office Action dated November 3, 2005 ("Office Action"), the Examiner objected to claim 4, rejected claims 1-2, 4-6, 13, 14, and 16-18 under 35 U.S.C. §102(b) as being anticipated by Shinohara, U.S. Patent No. 5,742,934 ("Shinohara"), rejected claims 7-8, 10, 19, and 20 under 35 U.S.C. §102(e) as being anticipated by Kahn et al., U.S. Patent No. 6,952,797 ("Kahn"), rejected claims 3 and 15 under 35 U.S.C. §102(e) as being anticipated by Biskup et al., U.S. Patent No. 6,751,757 ("Biskup"), rejected claim 9 under 35 U.S.C. §103(a) as being unpatentable over Kahn in view of Jackowski et al., U.S. Patent No. 4,809,273, and rejected claims 11 and 12 under 35 U.S.C. §103(a) as being unpatentable over Biskup in view of Kahn. Applicants' representative has amended claim 4 to address the Examiner's objection and respectfully traverses the above-mentioned 35 U.S.C. §102 and 35 U.S.C. §103 rejections.

First, having read the Examiner's rejections, Applicants' representative believes that the Examiner has failed to notice certain explanatory text in the current application that would facilitate the Examiner's understanding of certain of the claim terms. Beginning on line 5 of page 67, the current application states:

In ATA disk drives, as illustrated in Figure 36A, each sector of each track generally contains a data payload of 512 bytes. The sectors contain additional information, including a sector number and error-detection and error-correction information. This additional information is generally maintained and used by the disk-drive controller, and may not be externally accessible. *This additional information is not relevant to the current invention. Therefore, sectors will be discussed with respect to the number of bytes of data payload included in the sectors.* (emphasis added)

Thus, as clearly stated in the current application, and as well understood by those skilled in the art of disk drive design and manufacture, the term "sector length" refers to the size of the data payload of a sector, and not to the many other bits of information generally stored within sectors on disk drives that are generally created, manipulated, and accessed by the disk-drive controller, but that are generally inaccessible to external entities, including disk-array controllers, storage-shelf routers, and host computers. Those skilled in the art traditionally discuss physical sectors

having a size of 512 bytes, 520 bytes, or some other such number, well understanding that, as stored on disk, sectors include additional information. For example, a portion of an academic lecture published on the web page <http://www.cs.ualberta.ca/~macg/C429/Notes/RAID.pdf> describes disk-drive sectors as follows:

Sector formats vary between standards, but typically include:

- an initial gap
- a flag field for synchronizing the data separator (the chip that receives the analog signals from the heads and passes the results to the data formatter along with clock)
- address field in the cylinder / head / sector format, with CRC
- write space gap, to allow time for the heads to turn on when writing
- a flag field for synchronizing before reading data
- data bits - typically 512 to 4 Kbytes
- ECC field
- end of sector gap, used to accommodate small motor speed changes

The overhead for gaps, addresses and ECC is equivalent to about 40 to 100 bytes. Formatting a drive involves writing addresses, plus an arbitrary data pattern with the correct ECC.

Thus, the term "sector size" in the current application refers to data payload size, and not to any additional information stored within sectors, such as error correcting codes, parity bits, and other such non-data-payload information used by the disk-drive controller, but generally inaccessible to external entities. The above-quoted portion of the specification of the current application clearly defines "sector size" to mean data payload size, and those who design, manufacture, and use disk drives also refer to data-payload size when they discuss sector sizes.

Second, having read the Examiner's rejections, Applicants' representative believes that the Examiner has failed to properly understand the term "router." A router is not merely any kind of electronic device that transmits information. Instead, a router is a device that routes, or distributes, messages or data to some number of message or data receiving devices. For example, the Microsoft Press Computer Dictionary defines a router as:

An intermediary device on a communications network that expedites message delivery. On a single network linking many computers through a mesh of possible connections, a router receives transmitted messages and forwards them to their correct destinations over the most efficient available route.

In the current application, a storage router, an example of a general routing component, is described in the Summary of the Invention section, on page 27, as follows:

... is an integrated circuit implementing a storage-shelf router, used in combination with path controller cards and optionally with other storage-shelf routers *to interconnect SATA disks within a storage shelf or disk array to a high-bandwidth communications medium, such as an FC arbitrated loop.* (emphasis added)

Beginning on line 22 of page 31 in the current application, a storage-shelf router is described as follows:

As discussed above, with reference to Figures 8A-D, disk arrays may currently employ FC-compatible disk drives within storage shelves, each FC-compatible disk drive acting as an FC node on one or two FC arbitrated loops, or other FC fabric topologies, that interconnect the FC compatible disk drives with a disk-array controller. *By contrast, the storage-shelf router that represents, in part, one embodiment of the present invention serves as an intermediary communications hub, directly connected by point-to-point serial communications media to each disk drive within the storage shelf, and interconnected with the disk-array controller via one or more high-bandwidth communications media, such as fibre channel arbitrated loops.*

Overview

Figure 9 abstractly illustrates the storage-shelf router, representing one embodiment of the present invention, using the illustration convention employed for Figures 8A-D. *In Figure 9, disk-array controller 902 is linked via a LAN or fiber-optic communications medium 904 to one or more remote computer systems. The disk-array controller 902 is interconnected with a storage-shelf router 906 via an FC arbitrated loop 908. The storage-shelf router 906 is directly interconnected with each of the disk drives within a storage shelf 910-917 via separate point-to-point interconnects, such as interconnect 918.* (emphasis added)

Moreover, beginning on line 1 of page 36, various embodiments of a storage shelf are described, with reference to Figures 13A-C, as follows:

Figures 13A-C illustrate three different implementations of storage shelves using the storage-shelf router that represents, in part, one embodiment of the present invention. *In Figure 13A, a single storage-shelf router 1302 interconnects 16 SATA disk drives 1304-1319 with a disk-array controller via an FC arbitrated loop 1320.* In one embodiment, the storage-shelf router provides a maximum of 16 serial links, and can support interconnection of up to 16 SATA disk drives. The storage shelf shown in Figure 13A is not highly available, because it contains neither a

redundant storage-shelf router nor redundant serial links between one or more routers and each SATA disk drive.

By contrast, the storage-shelf implementation shown in Figure 13B is highly available. *In this storage shelf, two storage-shelf routers 1322 and 1324 are linked via point-to-point serial links to each of the 16 SATA disk drives 1326-1341.* (emphasis added)

In summary, a storage-shelf router is not simply an electronic device that transmits data to a single mass storage device, nor is a storage router a disk-controller or a disk-array controller. Instead, it is, in the embodiments described in the current application, a very complex, single integrated circuit that receives data packets and data-storage commands from one or more high-bandwidth communications media, such as a fibre channel communications medium, routes each command or data packet to one of multiple disk drives within a storage shelf, collects replies and data packets from the multiple disk drives within the storage shelf, and forwards the collected replies and data packets back to external entities via the high-bandwidth communications medium.

Third, the term "virtual disk formatting" does not refer to standard formatting carried out by disk-drive controllers or other mass-storage-device controllers. A disk-drive controller has full access to the physical mass-storage medium, can access and manipulate the extra bits stored on sectors, and can control seek, read, and write operations to access the physical medium according to a formatting convention. A disclosed embodiment of a storage-shelf router, by contrast, does not alter the formatting of disk drives within a storage shelf, but instead communicates with the individual disk-drive controllers of the disk drives within the storage shelf to issue access commands and to receive data and replies based on the formatting conventions employed by the disk drives. The storage-shelf router can carry out virtual disk formatting by, for example, translating commands from a disk-array controller that assumes 520-byte sectors to reside on disk drives to commands based on 512-byte sectors accepted by ATA disk drives. The disk drives are not reformatted. Thus, the formatting is "virtual." As described beginning on line 24 of page 67 of the current application:

The storage-shelf router that, in various embodiments, is the subject of the present invention allows economical ATA disk drives to be employed within storage shelves of a fiber-channel-based disk array. *However, certain currently available FC-based controllers may be*

implemented to interface exclusively with disk drives supporting 520-byte sectors. Although the manufacturer of an ATA or SATA-based storage shelf may elect to require currently-non-ATA-compatible disk-array controllers to be enhanced in order to interface to 512-byte-sector-containing ATA or SATA disk drives, a more feasible approach is to implement storage-shelf routers to support virtual disk formatting. Virtual disk formatting provides, to external entities such as disk-array controllers, the illusion of a storage shelf containing disk drives formatted to the FC-disk-drive, 520-byte-sector formatting convention, with the storage-shelf router or storage-shelf routers within the storage shelf handling the mapping of 520-byte-sector-based disk-access commands to the 512-byte-sector formatting employed by the ATA disk drives within the storage shelf.

Figures 37A-D illustrate the virtual-disk-formatting implementation for handling a 520-byte WRITE access by an external entity, such as a disk-array controller, to a storage-shelf-internal 512-byte-based disk drive. As shown in Figure 37A, external processing entities, such as disk-array controllers, view the disk to which a WRITE access is targeted as being formatted in 520-byte-sectors (3702 in Figure 37A), although the internal disk drive is actually formatted in 512-byte-sectors (3704 in Figure 37A). The storage-shelf router is responsible for maintaining a mapping, represented in Figure 37A by vertical arrows 3706-3710, between the logical 520-byte-sector-based formatting 3702 and the actual 512-byte-sector formatting 3704. (emphasis added)

With the above explanations, claim 1, provided below, is readily interpreted:

1. A virtual disk formatting system comprising:
 - a number of mass-storage devices having physical sectors of a first sector length; and
 - a routing component that provides a virtual disk interface to the mass-storage components by mapping access operations, received from external entities, directed to a virtual disk having virtual sectors of a second sector length to the number of mass-storage devices.

This claim is directed to the virtual formatting capability of a storage-shelf router that represents one embodiment of the present invention. The router component provides a virtual disk interface by mapping access operations directed to virtual sectors of a second sector length according to the virtual disk interface provided by the routing component to the sectors provided by the mass-storage devices. The mass-storage devices are not re-formatted. Again, as discussed above, actual sectors of an ATA drive contain additional bytes of information that are not part of the sector's data payload - but, as clearly stated in the current application, and as also discussed above, those additional bytes are irrelevant. The first and second sizes of sectors refer to the

data payload sizes of the sectors, as would be clearly understood by those familiar with disk drives and RAID controllers.

The Examiner's rejection of claims 1-2, 4-6, 13, 14, and 16-18 under 35 U.S.C. §102(b) as being anticipated by Shinohara is clearly unsupported. First, the Examiner proposes that (3) in Figure 1 is a router component - but, as clearly stated on lines 38-39 of column 3, (3) is a flash disk controller. It is not a router. This is clear from Figure 1, since the flash disk controller (3) communicates with only a single flash memory (4) and a single host computer (2). Routers necessarily involve communicating with multiple targets, in order to route commands or data received from the host computer to one of multiple receiving entities. Second, the Examiner proposes a mapping between virtual sectors having a second length to the number of mass-storage devices. Shinohara does not disclose multiple mass-storage devices, as discussed above. Most importantly - there is only a single sector size - namely, a sector size of 512 bytes. That is the sector size presumed by the host computer, and is also the sector size stored in the flash memory. The flash memory stores an additional 16 bytes of sector management information (see lines 63-67 of column 3) per sector, but that is not data payload, and is not, as clearly stated in the current application, to be considered as contributing to the sector size. There is no virtual sector discussed or disclosed in Shinohara. Instead, Shinohara discloses a simple flash-memory disk with 512-byte sectors. There is no routing component disclosed or discussed in Shinohara. Instead Shinohara merely discloses a simple flash-disk controller within a flash-disk card that provides access to 512-byte sectors to a host computer. Shinohara is unrelated to the currently claimed invention.

The Examiner's rejection of claims 7-8, 10, 19, and 20 under 35 U.S.C. §102(e) as being anticipated by Kahn is equally unsupported. In fact, it would seem that the Examiner has misinterpreted Kahn. For example, the Examiner suggests that the mass-storage devices (150 in Figure 1 of Kahn) have physical sectors having the length of a file block and an appended checksum. They have no such physical sector size. On lines 58-67 of Kahn, it is clearly stated that the disk drives (150) have 520-byte sectors. As also clearly stated in that passage, file blocks are not physical sectors, but are, instead, logical data units supported by a file system (generally a part of an operating system or, in other words, an entirely software implemented abstraction) that are mapped to sectors. Although the Examiner may feel that the

word "sector" can be freely interchanges with the word "block," they refer to entirely different entities. Sectors are the smallest unit of data that can be read or written from a mass-storage device. Blocks are data units of arbitrary length that are mapped, by a file system, to sectors. The mapping may not be to continuous sectors. File systems are not virtual disks, and file-system blocks are not virtual-disk sectors. The Examiner appears to state that item (220) in Figure 2 of Kahn is a virtual disk interface. Item 220 is a step in a method, in which a disk block is sent to a RAID controller. This step does not have, support, or provide virtual-disks or virtual-disk sectors. It is a data transmission step. The Examiner then appears to state that, referring to step 225 and non-existent step 226 of Figure 2, because the RAID controller computes a checksum and associates the checksum with the file-block data, that the RAID controller is somehow providing virtual sectors with a third length. File blocks are not sectors. Moreover, the file block and checksum have exactly the same length as the file block with checksum referred to by the Examiner as a first sector length. Finally, as discussed above, checksum bits and other non-data-payload information is not, in the current application, considered to contribute to a sector size. The term "sector size" refers only to data payload bytes.

Kahn's system merely provides for file-system-block-level checksums to be computed and appended to file-system blocks by a RAID controller, and the file-system blocks with checksums are then written across 8 disk sectors. There is no virtual disk interface provided in Kahn's system, but instead an implementation of a file system on top of a RAID controller. Most data storage systems are used to implement higher-level file systems, and file system blocks are generally significantly larger than disk sectors. File-system operations are directed to reading blocks from, and writing blocks to, files within file directories, and involve complex mapping of logical objects, including files and file blocks to mass-storage devices. Neither claim 7 nor the disclosed embodiments of the present invention mention file systems, implementation of file systems. Instead, claim 7 is directed to system in which a first virtual disk interface is provided by a routing component that allows an external entity to direct disk-access operations based on a second virtual sector size to the routing component. The routing component maps these access operations to equivalent access operations directed to a second, internal virtual disk interface that assumes a third virtual sector size, and finally maps the equivalent access operations to mass-

storage devices that provide physical sectors of a third sector size. In Kahn's system, by contrast, a file-system block of 4096 bytes is mapped to 8 520-byte disk sectors. There is no virtual disk interface disclosed or suggested by Kahn. Even were a file system regards as a virtual disk interface, which it clearly is not, the file-block checksum information is not relevant, and represents file-system management information that is generally not accessible by file-system users, and that is not considered to contribute to block size, according to the above-discussed explicit statement in the current application. Sector sizes in the current application refer only to data payload sizes, and not to checksum or other management information created, manipulated, and accessed by a controller. In other words, even were Kahn's file-blocks to be considered virtual sectors, there are only two block and sector sizes involved - a 4096-byte file block and a 520-byte disk sector. Kahn is, like Shinohara, unrelated to the invention claimed and disclosed in the current application.

Like Kahn and Shinohara, Biskup is unrelated to the invention disclosed and claimed in the current application. Biskup merely provides a different addressing of disk sectors so that every 33rd disk sector can be used to store a logical block address, CRC, and other such information. The disclosed addressing scheme is well illustrated in Figures 1 and 2 of Biskup. Figure 1, as discussed beginning on line 27 of column 4, shows that, in a conventional disk, consecutive sectors are addressed by consecutive logical block addresses. Figure 2 shows Biskup's different addressing scheme. In this scheme, every 33rd sector of the disk is skipped by the addressing scheme, so that every 33rd sector can be used by the disk controller to store management information, such as logical block address and CRC. This is succinctly stated in Biskup on lines 49-52:

In the illustrated embodiment, the LBAs for the system are mapped such that the use of every 33rd sector to store error checking information is transparent to the external system.


Biskup is even less related, if that is possible, to the currently disclosed and claimed invention than either of Kahn and Shinohara. There is but one sector size used in Biskup - namely the physical-sector size of the mass storage device. Although a disk or RAID controller can use additional sectors for storing management information, these additional sectors have the same size as the remaining sectors, and are not data

payload accessible or visible to external entities. Thus, the rejection of claims 3 and 15 as being anticipated by Biskup is unsubstantiated by anything disclosed in Biskup. Furthermore, the Examiner's arguments related to Biskup make no sense. The Examiner freely mixes numeric references to Figures 3 and 4 as analogs for claim elements, but, as clearly stated in Biskup, Figure 3 illustrates a conventional hard drive, and Figure 4 illustrates a drive manager (controller) that represents an embodiment of Biskup's claimed system. Furthermore, clusters are not sectors, as the Examiner seems to suggest, but are, instead, groups of sectors, as clearly shown in Figure 2. Nowhere in Biskup is there suggestion or mention of a sector size different from the sector size provided by the mass-storage device.

The Examiner's 35 U.S.C. §103 rejections of claims 9, 11, and 12 depend on Kahn and/or Biskup, and necessarily fail for the same reasons that the above 35 U.S.C. §102 must fail. Furthermore, since no the cited art is related to, anticipates, or makes obvious the independent claims, the dependent claims not specifically addressed in the above arguments cannot also be anticipated or made obvious by the cited art.

In Applicant's representative's opinion, all of the claims remaining in the current application are clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,
Joseph H. Steinmetz et al.
Olympic Patent Works PLLC


Robert W. Bergstrom
Registration No. 39,906

Enclosures:

Postcard
Transmittal in duplicate
Extension of Time
Check

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98194-0277
206.621.1933 telephone
206.621.5302 fax